OPTIMUM ACTUATOR SELECTION WITH A GENETIC ALGORITHM FOR AIRCRAFT CONTROL

JAMES L. ROGERS
NASA Langley Research Center

ABSTRACT:
The placement of actuators on a wing determines the control effectiveness of the airplane. One approach to placement maximizes the moments about the pitch, roll, and yaw axes, while minimizing the coupling. For example, the desired actuators produce a pure roll moment without at the same time causing much pitch or yaw. For a typical wing, there is a large set of candidate locations for placing actuators, resulting in a substantially larger number of combinations to examine in order to find an optimum placement satisfying the mission requirements and mission constraints. A genetic algorithm has been developed for finding the best placement for four actuators to produce an uncoupled pitch moment. The genetic algorithm has been extended to find the minimum number of actuators required to provide uncoupled pitch, roll, and yaw control. A simplified, untapered, unswept wing is the model for each application.

INTRODUCTION

Conventional control devices like flaps and ailerons have gaps between the wing and the control surface that contribute to leakage and protuberance drag. This can be a source of aerodynamic noise and increased observability. Flow control actuators potentially allow a seamless aircraft with no moving control surfaces, but rather hundreds of small ports capable of aerodynamically morphing the shape of the wing as needed. The cost and complexity of such a vehicle is obviously affected by the number and location of these ports (Scott et al. 1998).

The placement of the actuators on a wing determines the control effectiveness of the airplane. For the wings-leveler autopilot suggested by Scott et al, optimal placement means maximizing the moments about the pitch, roll, and yaw axes while minimizing the couplings among the moments. For a typical wing, there is a large set of candidate locations for placing actuators. The larger the set, the larger the possible combinations to examine in order to find an optimum subset to satisfy the mission requirements and mission constraints. Once engineering judgement has been used to identify potential regions for the actuators, a genetic algorithm (GA) is an excellent tool for determining the optimum placement.

The use of GA's has been instrumental in achieving good solutions to discrete optimization problems, such as the actuator placement problem, that have not been satisfactorily solved by other methods (Goldberg 1989). The discrete nature of the actuator placement problem has been recognized previously, and the GA approach has been successfully applied to solve the placement problem for interior noise control (Simpson and Hansen 1996).

The approach in this study differs in that the fitness of a population member is determined by calling PMARC (Ashby et al. 1988), a low-fidelity aerodynamic code for modeling complex three-dimensional geometries, to evaluate a multiobjective fitness function with several constraints. Given the placement of the actuators, the program returns the values for pitch, roll, and yaw. The model for this project is a simplified, untapered, unswept wing with 16 potential locations for actuators (Fig. 1).

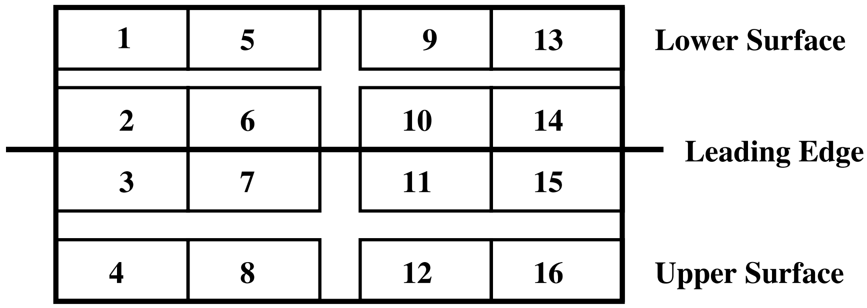

Figure 1. Wing model with 16 potential locations for actuators.

The results from this project demonstrate the effectiveness of applying a GA to optimize the placement of actuators for an aircraft design problem.

THE GENETIC ALGORITHM

This GA uses a direct representation of the order as a coding of a wing with n actuator locations. Each member of the population consists of a string of numbers where each position (1 through n) in the string contains a one or a zero, designating whether the actuator exists or not. For example, the member string [0 1 1 0 1 0 0 1 0 1 0 0 1 0 1 1] represents a wing with 16 possible actuator locations with actuators existing at locations 2, 3, 5, 8, 10, 13, 15, and 16. An initial population of members is randomly produced and evaluated. Successive populations are produced by the GA operations of selection, crossover, and mutation.

The selection operation determines those members of the population that survive to participate in the production of members of the next population. Selection is based on the value of the fitness function for the individual members. Members with better fitness levels tend to survive and are placed in the mating pool. Selection is accomplished by the tournament approach where two members are randomly selected from the parent pool and compared according to their fitness; the member with the best fitness is included in the mating pool.

The fitness of a member is determined by calling PMARC. The input for PMARC remains constant except for the array containing the actuator placements. The output from PMARC, the pitch, roll, and yaw moments, are used in determining the fitness of a member. Different fitness functions, described in detail below, are used in this project. PMARC requires about two minutes for the initial call (extra time for setting up some matrices) and one minute for each subsequent call on a Sun UltraSPARC™ workstation. PMARC is called to evaluate each member of the population, and there may be many generations of populations.

The crossover operation is the recombination of traits of the surviving members that have been placed in the mating pool, in the hope of producing a child with better fitness levels than its parents. This GA applies the single-point crossover technique as opposed to the uniform crossover method used by Simpson and Hansen. Single-point crossover is accomplished by randomly selecting two parent members from the mating pool and randomly selecting a crossover point. To create members for the next generation population, each location in the first parent member before and including the crossover point is copied to the first child, and each location after the crossover point is copied to the second child. Then the opposite locations from the second parent are copied to each child.

The mutation operation prevents the search of the design space from becoming too narrow. After the production of a child population, mutation randomizes small parts of the resulting members, with a very low probability that any given member location will be affected. Mutation is accomplished by polling each location in the member. A random number generator, along with a user-defined mutation parameter (default is .01), is used to determine if that location is to be mutated. If mutation occurs and there is a one in the location, it is made a zero, and vice versa.

SINGLE OBJECTIVE APPLICATION

For each application, the wing model was designed with 16 potential locations for actuators (Fig. 1). The problem statement for the first application of a GA to actuator placement was, "Given 16 actuator locations, find the best placement for four actuators to maximize pitch-down with very little or no roll or yaw moments." The population size for this application was 100. The fitness function maximized the pitch-down. Three constraints were added to penalize the fitness function by adding a positive number for each violation.

(1) if the absolute value of the roll is greater than .001, add 1
(2) if the absolute value of the yaw is greater than .001, add 1
(3) if the number of actuators is not equal to 4, add 10

The GA found several configurations that satisfy the constraints (Fig. 2). In the figure, the shaded locations indicate where actuators need to be placed. The optimum fitness was -0.078 for a case with four actuators on the top surface of the wing, near the leading edge (Fig. 2c).

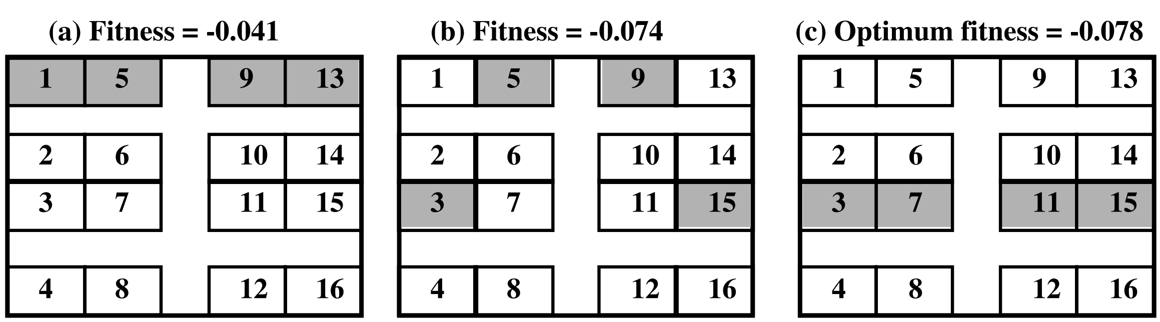

Figure 2. Actuator configurations for pitch-down.

MULTIOBJECTIVE FUNCTION

The problem statement for the second application of a GA to actuator placement is, "Given 16 actuator locations, find the minimum number of actuators required to provide uncoupled pitch, roll, and yaw moments." The problem was divided into three distinct subproblems (one each for uncoupled pitch, uncoupled roll, and uncoupled yaw), each with a separate population. The fitness function for each subproblem found the minimum number of actuators required for that particular uncoupled maneuver. The value of the moment was not a part of the fitness function for this application. There was also a composite fitness function that found the minimum number of actuators required to accomplish all three uncoupled maneuvers. There were five possible constraints for each subproblem and each added a positive number when violated. For example, the constraints for the pitch subproblem were:

(1) if the absolute value of the roll is greater than .001, add 20
(2) if the absolute value of the yaw is greater than .001, add 20
(3) if the number of actuators is less than some minimum number, add 80
(4) if the PMARC code does not converge, add 80
(5) if the absolute value of the pitch is less than .001, add 20

The penalty of 20 for constraints (1), (2) and (5) was chosen because the maximum number of actuators that could be applied was 16 and a penalty was needed that made the fitness more than 16. Constraint (3) was added so that engineers could specify a minimum number of actuators for safety reasons. (The initial minimum of 4 was later reduced to 2.) Constraint (4) was added to take advantage of convergence information from the PMARC convergence parameter. Constraints (3) and (4) were more heavily penalized than (1) and (2) to move the search away from these members more rapidly. Constraint (5) was not a part of the initial problem. Initially, the penalties were cumulative.

The population size for this application was 100 and there were different populations for pitch, roll, and yaw. The GA iterated through the 100 members of each subproblem, and evaluated the fitness of each individual member. If the constraints were satisfied, the strings of the individual members were combined with an "OR" function to form a composite member. The fitness of the composite member was the sum of the actuators that would be available for each of the uncoupled maneuvers.

Examining the composite fitness one member at a time proved to be inefficient. For example, in one pass through the population, the two members selected for pitch and yaw each had an excellent fitness of 4, however, the one selected for roll had a poor fitness of 10. This resulted in a composite fitness of 16. Another pass found members with the following fitnesses: pitch = 10, roll = 4, yaw = 10, and composite = 18. If the pitch and yaw strings from the first pass were combined with the roll string from the second pass, then the composite fitness would be 9. Thus, arrays were added to save members of each subproblem that do not violate any constraints, and all possible good combinations of strings from the subproblems can now be evaluated to find the best combination for composite fitness.

Another problem that occurred in this application was the lack of a test for the individual moments. Members that had no pitch, roll, and yaw were accepted as valid. Therefore, constraint (5) was added to each subproblem.

Because absolute values were used in the evaluation, the GA could find pitchup or pitch-down, roll left or roll right, and yaw left or yaw right. In this application, the GA found pitchup (Fig. 3a), roll left (Fig. 3b), and yaw right (Fig. 3c). The results in Figure 3 were obtained after 18 generations. The composite configuration placed actuators at locations 4, 6-11, 13, and 16.

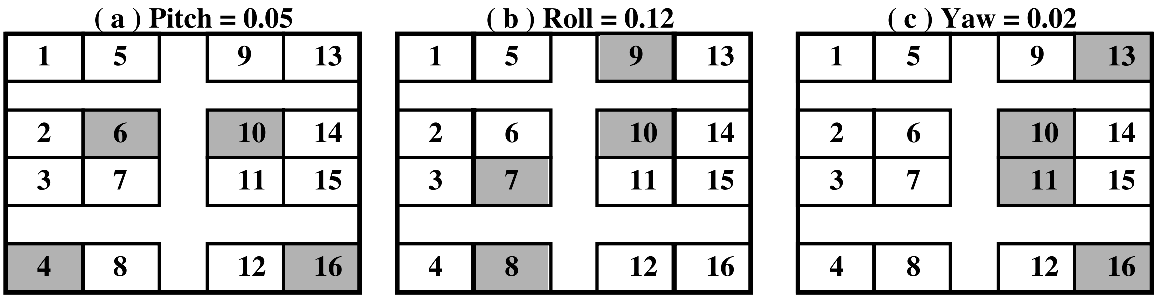

Figure 3. Four-actuator placement with uncoupled moments.

These results were then shown to engineers from NASA Langley's Dynamics and Controls Branch. Their suggestion to relax constraint (3) to allow a minimum of two actuators per subproblem was incorporated into the program. In addition, the penalties were no longer cumulative, as that seemed to slow down the convergence process. In this application, the GA found pitchup (Fig. 4a), roll right (Fig. 4b), and yaw right (Fig. 4c). The results, obtained after 13 generations, from these changes are shown in Figure 5. The composite configuration placed actuators at locations 3, 4, 13, 14, and 16.

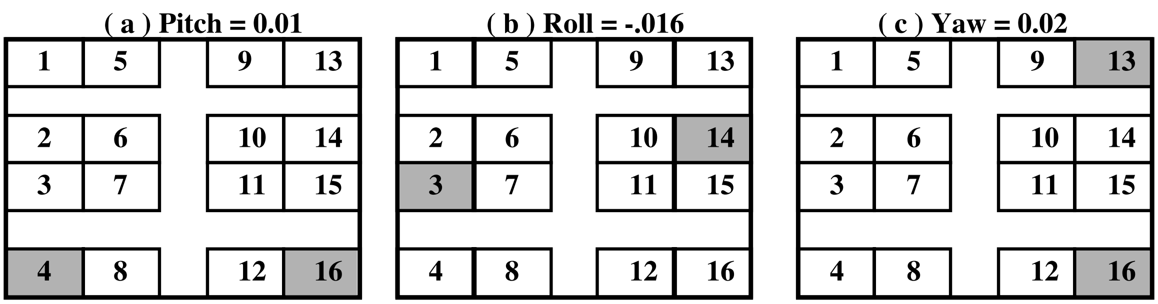

Figure 4. Two-actuator placement with uncoupled moments.

The wing model is symmetric, so this information can be used to determine a composite configuration for all six uncoupled maneuvers for both the four- and two-actuator placement configurations (Figs. 5a and 5b, respectively). Future applications will use the same model, but will increase in complexity, with additional locations for potential actuators as well as the examination of coupled maneuvers.

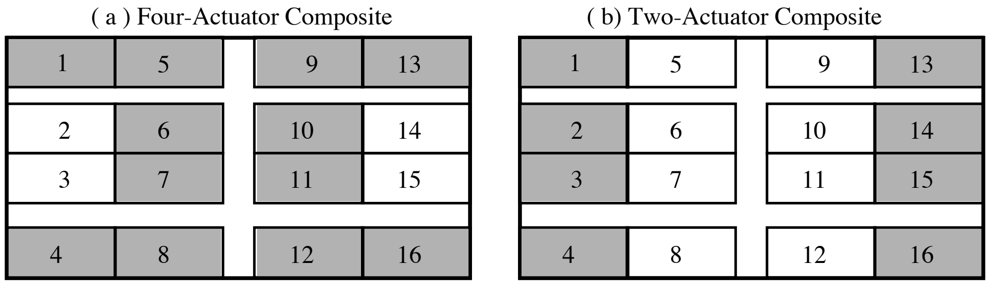

Figure 5. Symmetric composite for four- and two-actuator placement configurations.

CONCLUSIONS

A genetic algorithm has been successfully applied to the placement of actuators for a simplified, untapered, unswept wing. This wing model has 16 potential locations for actuators. A low-fidelity aerodynamic code is used to determine the fitness of each member of the population. The first application contains a single objective function which finds the best placement for four actuators to maximize pitch-down with three constraints. The second application uses the same wing model, but contains a multiobjective function which finds the minimum number of actuators required to provide uncoupled pitch, roll, and yaw control. The problem is divided into three distinct subproblems, each with its own population, fitness function, and constraints. A composite fitness function finds the minimum number of actuators required to accomplish all three uncoupled maneuvers. In this application, the genetic algorithm finds the optimum placement for a minimum of four (then two) actuators in each subproblem for the three uncoupled maneuvers. Wing symmetry is used to determine a composite configuration for all six uncoupled maneuvers.

Acknowledgement. The author would like to acknowledge Dr. William A. Crossley of Purdue University for his development in the PMARC model and consultation.

REFERENCES

Ashby, D. L., Dudley, M. R., and Iguchi, S. K. (1988). *Development and Validation of an Advanced Low-Order Panel Method.* NASA TM 101024.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Co., New York.

Scott, M. A., Montgomery, R. C., and Weston, R. P. (1998). Subsonic Maneuvering Effectiveness of High Performance Aircraft Which Employ Quasi-Static Shape Change Device. SPIE's 5th Annual International Symposium on Smart Structures and Materials, San Diego, pp.223-233.

Simpson, M. T., and Hansen, C. H. (1996). Use of Genetic Algorithms to Optimize Vibration Actuator Placement for Active Control of Harmonic Interior Noise in a Cylinder with Floor Structure. *Noise Control Engineering Journal*, Vol. 44, No. 4, pp. 169-184.